SUPPORTING INFORMATION

Catalytic Oxygenation-Mediated Extraction as a Facile and Green Way to Analyze Volatile Solutes

Chamarthi Maheswar Raju[1], Kai-Chiang Yu[1], Chun-Pei Shih[1], Decibel P. Elpa[1,2], Gurpur Rakesh D. Prabhu[1], Pawel L. Urban[1,3]*

[1] *Department of Chemistry, National Tsing Hua University*
101, Section 2, Kuang-Fu Rd., Hsinchu, 30013, Taiwan
[2] *Department of Applied Chemistry, National Yang Ming Chiao Tung University*
1001 University Road, Hsinchu 30010, Taiwan
[3] *Frontier Research Center on Fundamental and Applied Sciences of Matters,*
National Tsing Hua University
101, Section 2, Kuang-Fu Rd., Hsinchu, 30013, Taiwan

* Corresponding author:
Prof. P.L. Urban
E-mail: urban@mx.nthu.edu.tw

Contents:

ADDITIONAL EXPERIMENTAL DETAILS

Samples

Lemon drink (ingredients: lemon juice mixed with condensed milk, vitamin C, carboxymethyl cellulose, water), orange juice (ingredients: 100% orange pulp, sugar, vitamin C, water), apple flavor milk (ingredients: condensed milk, sugar, artificial apple flavors, apple pulp), strawberry-flavored yogurt (ingredients: concentrated strawberry pulp, sugar, natural yogurt), American whiskey (45% v/v), Scotch whiskey (40% v/v), and potatoes were purchased from local supermarkets (Hsinchu City, Taiwan). Urine specimen was collected from a healthy volunteer (male) before analysis. Wastewater was collected from the Zhubei area (Hsinchu County, Taiwan), and stored at a temperature of 4 °C for 44 days before the analysis by GC-MS and 46 days before analysis by APCI-MS. Thus, it is expected that the storage contributed to the final composition of this matrix.

For analysis of real samples by COME with GC-MS, 50 μL aliquots of the samples (lemon drink, and orange juice), were mixed with 2010 μL of 50 mM potassium dihydrogen phosphate buffer adjusted to pH 7.0 with sodium hydroxide and 360 μL of 2500 U mL^{-1} catalase stock solution (dilution factor: 60×). In the case of apple flavor milk, the volumes of sample and buffer were 100 μL and 1960 μL, respectively (dilution factor: 30×). In the case of stored waste water, the volumes of sample and buffer were 500 μL and 1560 μL, respectively (dilution factor: 6×). Subsequently, 360 μL of 2500 U mL^{-1} added before extraction and 580 μL of ~ 35% hydrogen peroxide solution was infused during the oxygenation process. Thus, the final volume of the mixture was 3 mL.

For analysis of real samples by COME with APCI-MS, 25 μL of lemon drink was mixed with 2035 μL of 50 mM potassium dihydrogen phosphate buffer adjusted to pH 7.0 with sodium hydroxide and 360 μL of 2500 U mL^{-1} catalase stock solution (dilution factor: 120×). In the case of orange juice, the volumes of sample and buffer were 50 μL and 2010 μL, respectively (dilution factor: 60×). In the case of apple flavor milk, the volumes of sample and buffer were 100 μL and 1960 μL, respectively (dilution factor: 30×). In the case of stored waste water, the volumes of sample and buffer were 500 μL and 1560 μL, respectively (dilution factor: 6×). Subsequently, 360 μL of 2500 U mL^{-1} added before extraction and 580 μL of ~ 35% hydrogen peroxide solution was infused during the oxygenation process. Thus, the final volume of the mixture was 3 mL.

For analysis of real samples by COME with APCI-IMS, 100 μL aliquots of the samples (American whiskey and Scotch whiskey) were mixed with 1960 μL of 50 mM potassium dihydrogen phosphate buffer adjusted to pH 7.0 with sodium hydroxide and 360 μL of 2500 U mL^{-1} catalase stock solution (dilution factor: 30×). In the case of strawberry-flavored yogurt, the volumes of sample and buffer were 500 μL and 1560 μL, respectively (dilution factor: 6×). In the case of urine, the volumes of sample and buffer were 1500 μL and 560 μL, respectively (dilution factor: 2×). Subsequently, 360 μL of 2500 U mL^{-1} added before extraction and 580 μL of ~ 35% hydrogen peroxide solution was infused during the oxygenation process. Thus, the final volume of the mixture was 3 mL.

Gas chromatography – mass spectrometry

In the first variant, the COME apparatus (**Figure 1A**) was coupled with a gas chromatograph (Trace GC Ultra; Thermo Fisher Scientific, Waltham, MA, USA) hyphenated to a single quadrupole mass spectrometer (ISQ; Thermo Fisher Scientific, Waltham, MA, USA). The GC column was SPB-5 capillary column (5% diphenyl / 95% dimethyl siloxane; length, 30 m; ID, 0.25 mm; film thickness, 0.25 μm; Supelco, Bellefonte, PA, USA). The analysis conditions were as follows: injection volume, 1 μL (during initial tests); injector temperature, 200 °C; split ratio, 12; carrier gas, helium (99.999%); flow rate, 0.8 mL min^{-1}. The GC temperature program was: initial temperature at 40 °C, then ramped up at a rate of 10 °C min^{-1} until the temperature reached 200 °C. The run time of the GC method was 16 min. Ions were generated by electron ionization (EI) at 70 eV. Real sample analyses were performed in full scan mode within the *m/z* range from 40 to 300. Quantitative analyses were conducted in selected ion monitoring (SIM) mode as follows: start acquisition at 4.00 min, *m/z* 43,56, 69, 93, and 121; from 7.92 min, *m/z* 68, and 93; and finally from 8.45 to 16.00 min, *m/z* 93, 121, and 136. After the data acquisition, the extracted ion current (EIC) profiles were exported for the following ions: *m/z* 93 (BP, BM, and GT); *m/z* 68 (LM) and *m/z* 56 [HA (internal standard)]. Afterwards the data were exported to Excel format, and imported to RStudio software (ver. 1.3.1093; RStudio, Boston, MA, USA) for plotting and signal-to-noise calculations. Xcalibur software (ver. 2.1.0 SP1.1160; Thermo Fisher Scientific) was used for peak area determination.

Atmospheric pressure chemical ionization mass spectrometry

In the second variant, the COME apparatus (**Figure 1C**) was coupled with a triple quadrupole mass spectrometer (LCMS-8030; Shimadzu, Tokyo, Japan) fitted with a corona discharge APCI source. The settings of the mass spectrometer were: nebulizing gas (nitrogen) flow rate, 2.0 L min^{-1}; drying gas (nitrogen) flow rate, 10.0 L min^{-1}; corona needle voltage, 4.5 kV; desolvation line temperature, 250 °C; heated block temperature, 250 °C; run time, 1 min. Positive-ion APCI scan mode (*m/z* 40-300) was used for the analyses of real samples (lemon drink, orange juice, apple flavor milk, and the stored wastewater). Positive-ion APCI multiple reaction monitoring (MRM) mode was used for the analyte (LM; *m/z* 137→81, 137→95) and the internal standard (HA; *m/z* 145→117, 145→61). The collision induced dissociation (CID) gas (argon, 99.99%) was supplied at a pressure of 230 kPa. The collision voltages were set to -20 V for LM and -10V for HA. After the data acquisition, EICs were exported as ASCII files and plotted using R software. Average ion intensities of EICs [*m/z* 137→81 (LM), *m/z* 145→117 (HA)] from 0.4 to 0.8 min were used to plot the calibration curve.

Ion-mobility spectrometry

In the third variant, the COME apparatus (**Figure 1E**) was coupled with an ion-mobility spectrometer (OEM-IMS module; G.A.S., Dortmund, Germany) fitted with a tritium (^3H) APCI source. A 13-cm PTFE tube (length, 13 cm; ID, 0.3 mm; OD, 1.58 mm; cat. no., 58702; Supelco, Bellefonte, PA, USA) was used to connect the extraction chamber to valve 1 [including 4-cm silicone tubing section (ID, 0.81 mm; OD, 2.52 mm)]. Further, 6-cm PTFE tube (length, 13 cm; ID, 0.3 mm; OD, 1.58 mm; cat. no., 58702; Supelco, Bellefonte, PA, USA) was used to connect valve 1 to the sample inlet of IMS *via* stainless steel tube fitting union (1/16 inch tube OD; Solon, OH, USA). The IMS parameters were as follows: drift voltage polarity, positive; injection pulse

width, 150 μs; drift voltage, 240 V; drift gas flow rate, 157 mL min^{-1}; recording time, 60 s; repetition rate, 20 ms; number of averaged spectra, 12; blocking voltage, 120 V; and drift tube temperature, 70 °C. After the recording time, data were exported as CSV files and processed and plotted using R software. Average signals from 20 to 60 s recording time were chosen for ion-mobility spectra.

ADDITIONAL TABLES

Table S1. Calibration equations and LODs for four test compounds were analyzed by GC-MS in SIM mode (calibration values, $n = 2$; blank values for LOD calculation, $n = 3$). Six calibration levels ranging from 5×10^{-10} M to 1×10^{-7} M. Internal standard HA (1×10^{-8} M; $m/z = 56$) was used to compensate for technical variability. The concentration values refer to the final 3-mL reaction mixture. For the calibration plots, please refer to **Figure S5**.

No.	Compound	t_R / min	m/z	Calibration equation (C / M)	R^2	Range / M	LOD / M (SD)
1	BP	7.39	93	$A_{STD}/A_{ISTD} = [(1.46\pm0.04)\times10^{7}]C_{STD} + [(4.32\pm21.6)\times10^{-3}]$	0.989	$5\times10^{-10} – 1\times10^{-7}$	$(4.53\pm1.18)\times10^{-11}$
2	BM	7.52	93	$A_{STD}/A_{ISTD} = [(5.41\pm0.19)\times10^{6}]C_{STD} + [(-3.61\pm8.52)\times10^{-3}]$	0.988	$5\times10^{-10} – 1\times10^{-7}$	$(1.23\pm0.70)\times10^{-10}$
3	LM	8.17	68	$A_{STD}/A_{ISTD} = [(5.74\pm0.12)\times10^{6}]C_{STD} + [(5.26\pm5.51)\times10^{-3}]$	0.995	$5\times10^{-10} – 1\times10^{-7}$	$(4.80\pm0.27)\times10^{-10}$
4	GT	8.63	93	$A_{STD}/A_{ISTD} = [(4.32\pm0.18)\times10^{6}]C_{STD} + [(-5.39\pm8.70)\times10^{-3}]$	0.981	$5\times10^{-10} – 1\times10^{-7}$	$(1.87\pm0.36)\times10^{-10}$

Table S2. Repeatability and reproducibility of selected compounds (1×10^{-8} M) analyzed by GC-MS based on SIM mode. HA (1×10^{-8} M, *m/z* = 56) was used as an internal standard for all test compounds (BP, BM, LM, and GT). RSDs were calculated based on the ratio of analyte peak area and internal standard area.

No.	Compound	t_R / min	*m/z*	Repeatability (RSD) / % *n* = 10	Reproducibility (RSD) / % *n* = 18 (6 days)
1	BP	7.39	93	12.1	11.6
2	BM	7.52	93	12.5	12.9
3	LM	8.17	68	15.1	13.4
4	GT	8.63	93	11.2	11.8

Table S3. Recoveries of four test compounds (BP, BM, LM, and GT) present in the lemon drink were determined using GC-MS (SIM mode; $n = 2$). The sample was diluted 200× to analyze BP, BM, and GT. In the case of the LM analysis sample was diluted 5000× to fit the concentration of analyte in the calibration range. HA (1×10^{-8} M, $m/z = 56$) was used as an internal standard.

Sample matrix	Compound	t_R / min	m/z	Result of quant. analysis before spiking (diluted sample) / M	Spiked concentration / M	Result of quant. analysis after spiking / M	Recovery / %
lemon drink	BP	7.39	93	8.44×10^{-8}	5.00×10^{-8}	1.22×10^{-7}	74.6
lemon drink	BM	7.52	93	1.89×10^{-8}	5.00×10^{-8}	5.48×10^{-8}	71.8
lemon drink	LM	8.17	68	2.66×10^{-8}	5.00×10^{-8}	6.62×10^{-8}	79.2
lemon drink	GT	8.63	93	4.03×10^{-8}	5.00×10^{-8}	8.46×10^{-8}	88.6

Table S4. Comparison of different sampling/extraction techniques (*cf.* **Figure S12**). Signal-to-noise (*S/N*) ratios of 2 replicates (SIM mode) were averaged (spread values reported). Concentrations of the four test compounds (BP, BM, LM, and GT) were 1×10^{-8} M.

No.	Method	*S/N* (BP)	*S/N* (BM)	*S/N* (LM)	*S/N* (GT)
1	HS-SDME (room temperature, 15 min, stirring 200 RPM)	35.5 ± 4.7	21.7 ± 2.7	29.0 ± 2.5	14.1 ± 1.8
2	HS-SDME (40 °C, 5 min, stirring 200 RPM)	14.7 ± 1.3	12.8 ± 1.0	14.3 ± 1.0	11.6 ± 1.7
3	HS (room temperature, 15 min, stirring 200 RPM)	215 ± 17	81.5 ± 3.0	150 ± 2.3	91 ± 15
4	HS (40 °C, 5 min, stirring 200 RPM)	232 ± 20	90.1 ± 1.7	162 ± 12	110 ± 13
5	COME	415 ± 26	168 ± 12	319 ± 20	182 ± 7.3
6	HS-SPME (room temperature, 15 min, stirring 200 RPM)	3719 ± 662	2481 ± 273	6081 ± 235	4134 ± 163
7	HS-SPME (40 °C, 5 min, stirring 200 RPM)	1681 ± 160	871 ± 98	1344 ± 76	1156 ± 105
8	COME + HS-SPME (room temperature, 5 min)	3618 ± 227	2573 ± 542	6050 ± 357	4424 ± 402
9	COME + HS-SPME (room temperature, 15 min)	32956 ± 2409	21073 ± 4948	34932 ± 813	44706 ± 921

Table S5. Comparison of COME variants using alternative catalysts (*cf.* **Figure S14**). Signal-to-noise (*S/N*) ratios of 2 replicates (SIM mode) were averaged (spread values reported). Concentrations of the four test compounds (BP, BM, LM, and GT) were 1×10^{-8} M.

No.	Conditions	*S/N* (BP)	*S/N* (BM)	*S/N* (LM)	*S/N* (GT)
1	Pure catalase (300 U mL^{-1})	415 ± 31	168 ± 12	319 ± 20	182 ± 8.5
2	Potato pulp (1.25 g)	94.2 ± 10.2	32.3 ± 1.9	87.6 ± 8.2	34.5 ± 4.1
3	Manganese(IV) dioxide (125 mg)	399 ± 23	166 ± 25	289 ± 27	167 ± 14

ADDITIONAL FIGURES

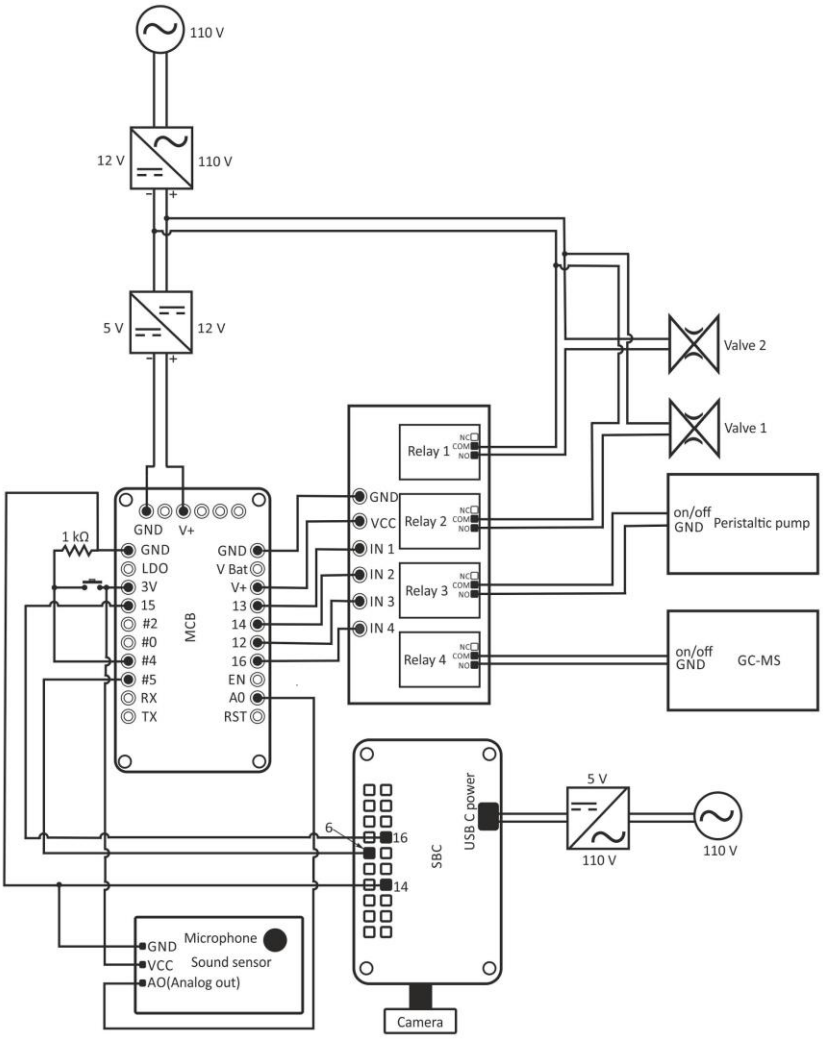

Figure S1. Scheme of the electronic system used to control the extraction process. Note that a monitor is optionally connected to the SBC.

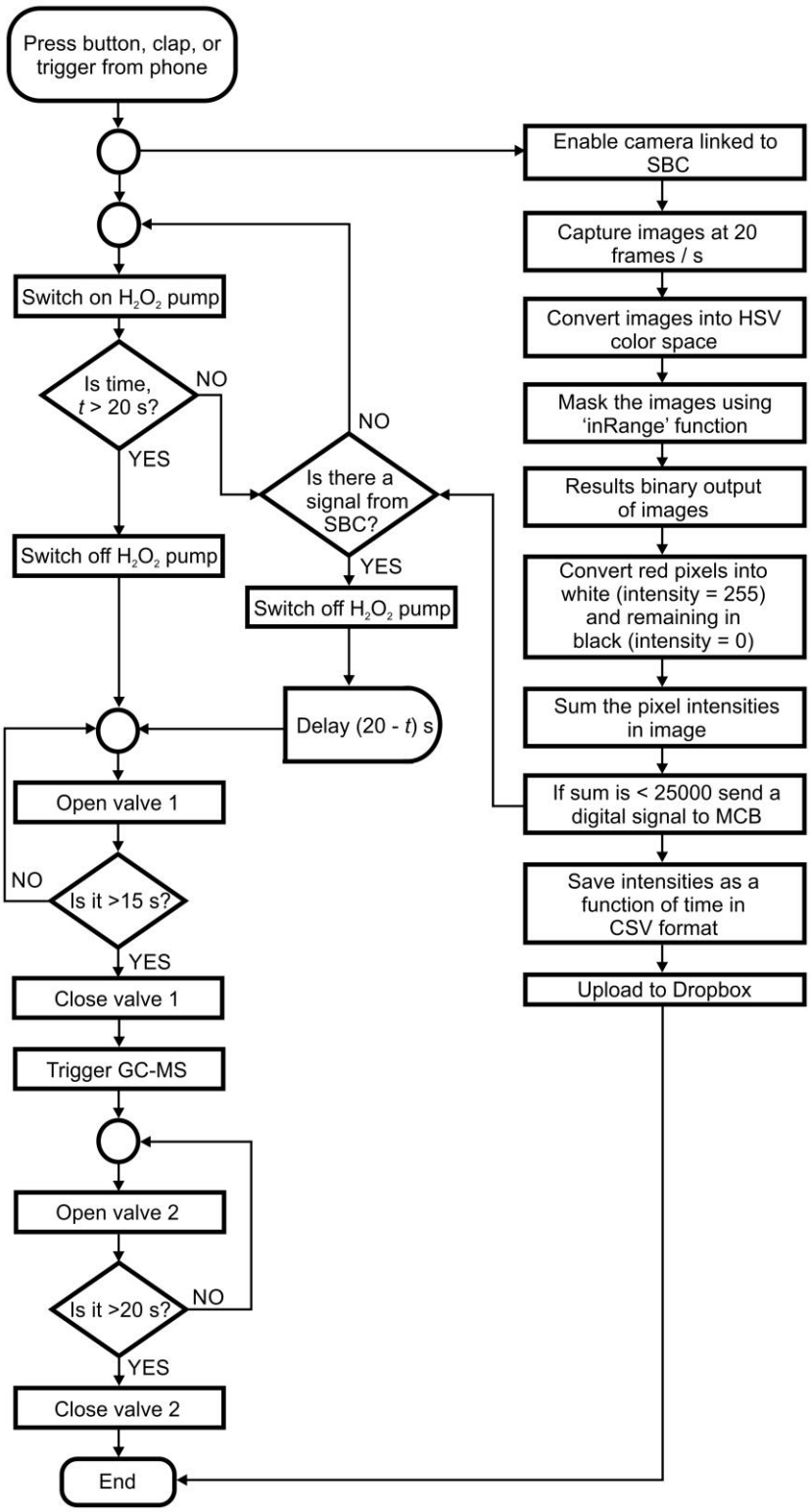

Figure S2. Workflow of the program used to control the extraction process.

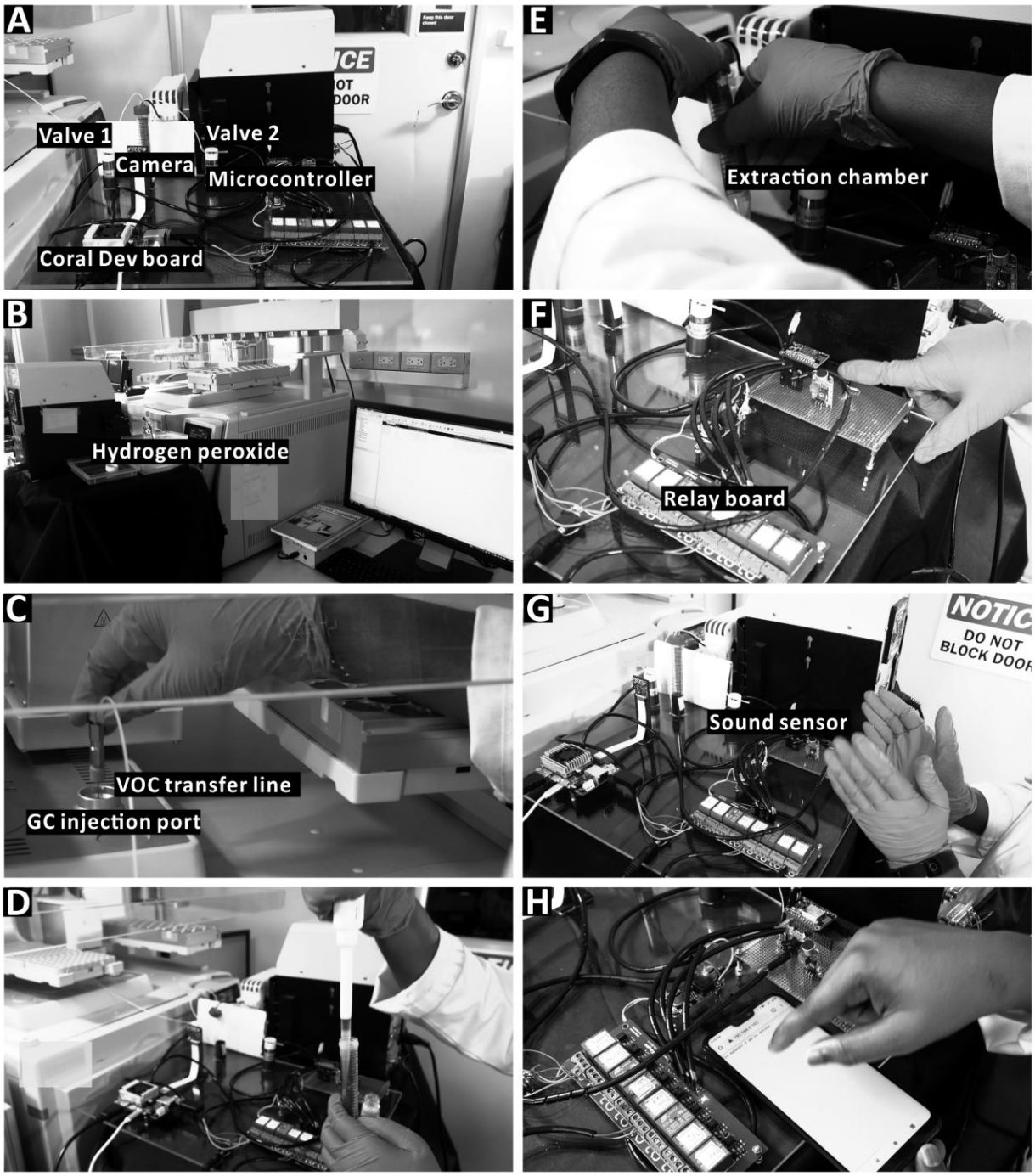

Figure S3. Photos of the entire system. (A) left side view; (B) right side view; (C) COME apparatus connected to GC injection port; (D) catalase added before extraction; (E) extraction chamber closed tightly; (F) extraction triggered by 'start' button; (G) extraction triggered by clapping sound; (H) extraction triggered by smartphone. For a related video, see **Movie S1**.

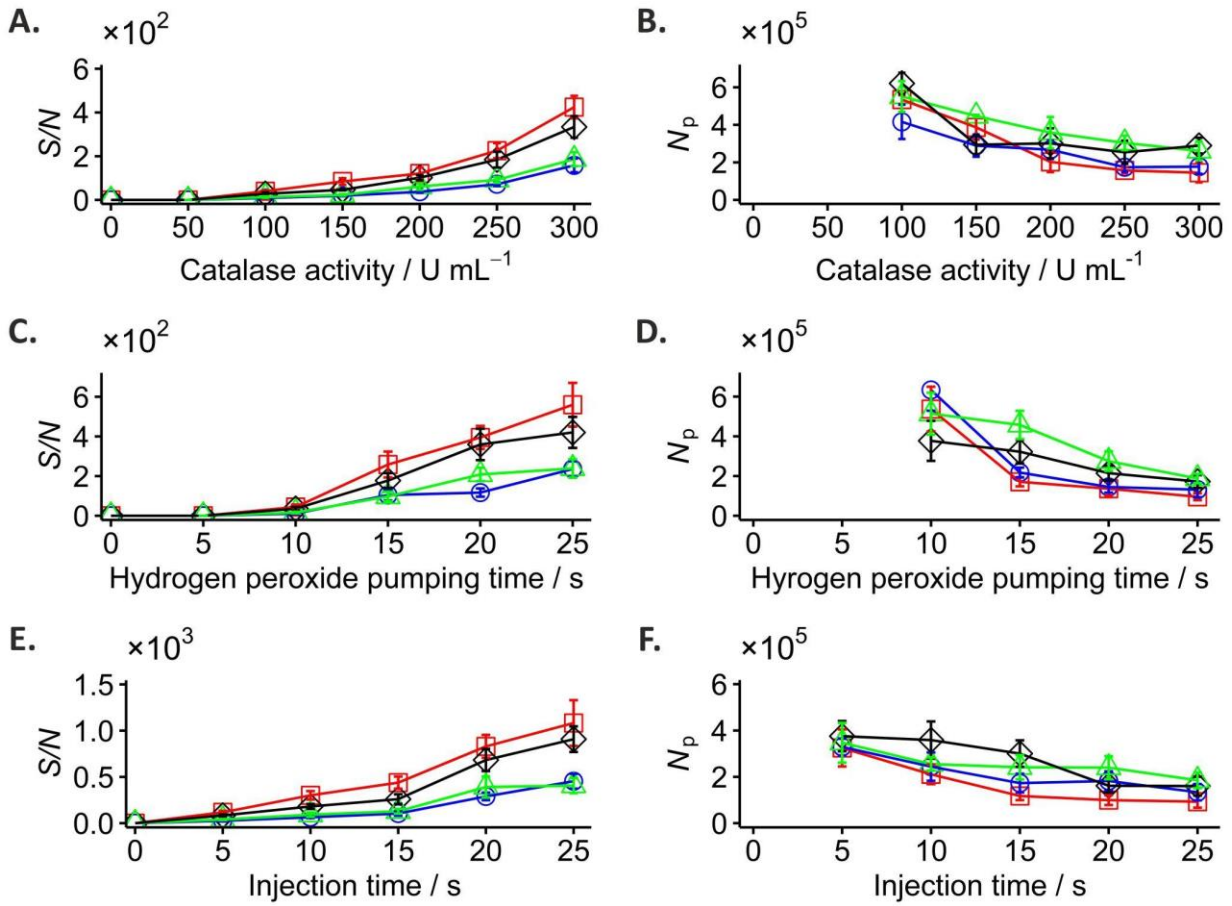

Figure S4. Optimization of extraction conditions. (A) *S/N* ratio *vs.* catalase activity; (B) number of theoretical plates (N_P) *vs.* catalase activity; (C) *S/N* ratio *vs.* hydrogen peroxide pumping time; (D) number of theoretical plates (N_P) *vs.* hydrogen peroxide pumping time; (E) *S/N* ratio *vs.* injection time; (F) number of theoretical plates (N_P) *vs.* injection time. Symbols: (square, red line) BP; (diamond, black line) BM; (triangle, green line) LM; (circle, blue line) GT. Concentrations of the four test compounds (BP, BM, LM, and GT) were 1×10^{-8} M. Default conditions: catalase activity, 300 U mL^{-1}; hydrogen peroxide pumping time, 20 s; injection time, 15 s; injector temperature, 200 °C; split ratio, 12; carrier gas, helium (99.999%); flow rate, 0.8 mL min^{-1}. The default GC temperature program was: initial temperature at 40 °C, then ramped up at a rate of 10 °C min^{-1} until the temperature reached 200 °C. The run time of the GC method was 16 min. Ions were generated by EI at 70 eV. The SIM mode EICs [*m/z* 93 (BP, BM, GT); *m/z* 68 (LM)] were used in data analysis for all test compounds.

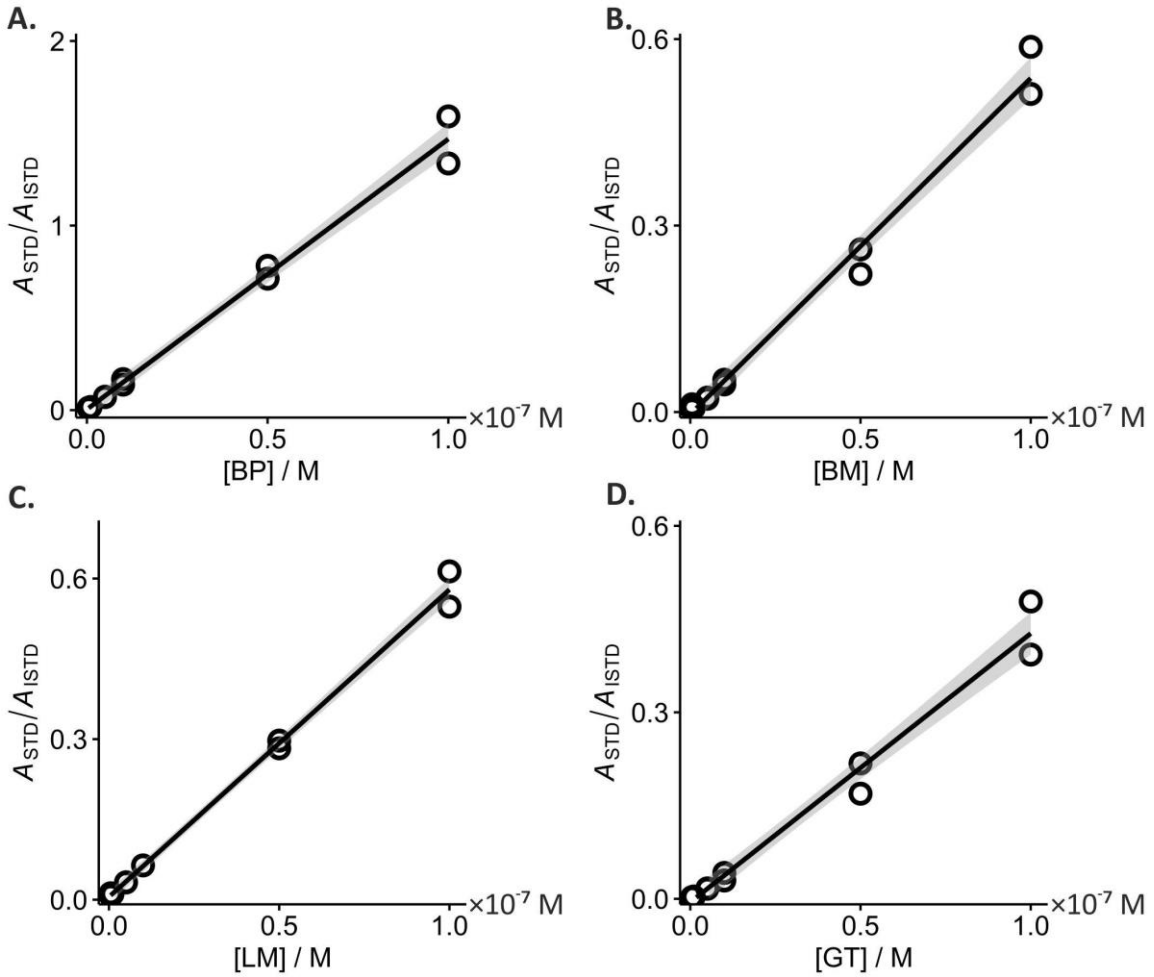

Figure S5. Calibration plots for four test compounds analyzed by GC-MS in SIM mode (n = 2). Panels: (A) BP; (B) BM; (C) LM; (D) GT. HA (1×10^{-8} M) was used as an internal standard. The concentration values refer to the final 3-mL reaction mixture. For the calibration equations, please refer to **Table S1**. Conditions: catalase activity, 300 U mL^{-1}; hydrogen peroxide pumping time, 20 s; injection time, 15 s; injector temperature, 200 °C; split ratio, 12; carrier gas (helium, 99.999%) flow rate, 0.8 mL min^{-1}. The default GC temperature program was: initial temperature at 40 °C, then ramped up at a rate of 10 °C min^{-1} until the temperature reached 200 °C. The run time of the GC method was 16 min. Ions were generated by EI at 70 eV. The SIM mode EICs [*m/z* 93 (BP, BM, GT); *m/z* 68 (LM)] were used in data analysis for all test compounds (BP, BM, LM, and GT).

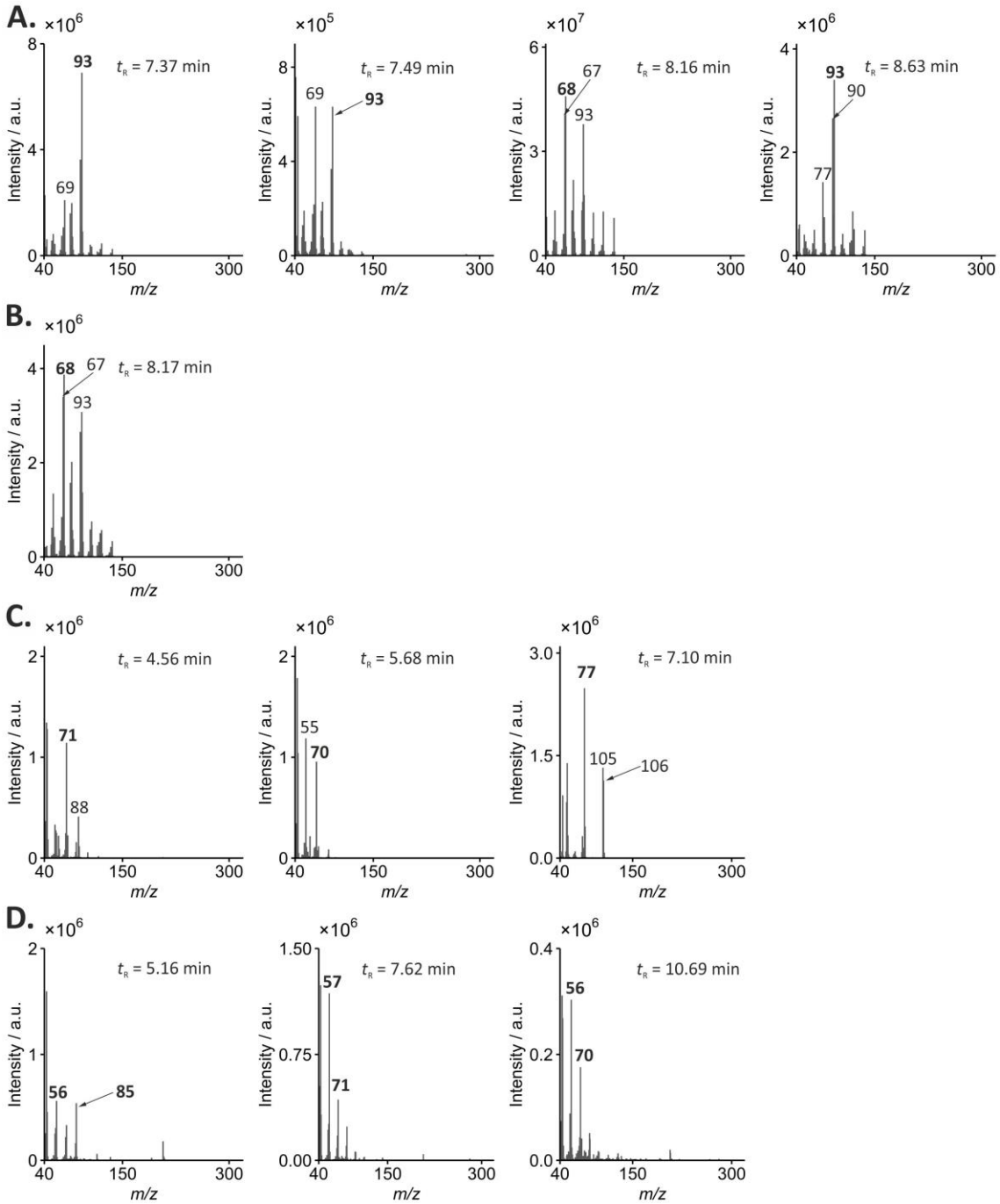

Figure S6. Analysis of VOCs by COME in conjunction with GC-MS (*cf*. **Figure 1A,B**). The *m/z* values chosen for plotting the chromatograms in **Figure 3A** are indicated with bold font. In lemon drink (A), BP (t_R = 7.37 min), BM (t_R = 7.49 min), LM (t_R = 8.16 min), and GT (t_R= 8.63 min) were detected. In orange juice (B), LM (t_R = 8.17 min) was detected. In apple flavor drink (C), EB (t_R = 4.56 min), MB (t_R = 5.68 min), and BA (t_R = 7.10 min) were detected. In the stored wastewater (D), OT (t_R = 5.16 min), DC (t_R = 7.62 min), and DD (t_R = 10.69 min) were detected.

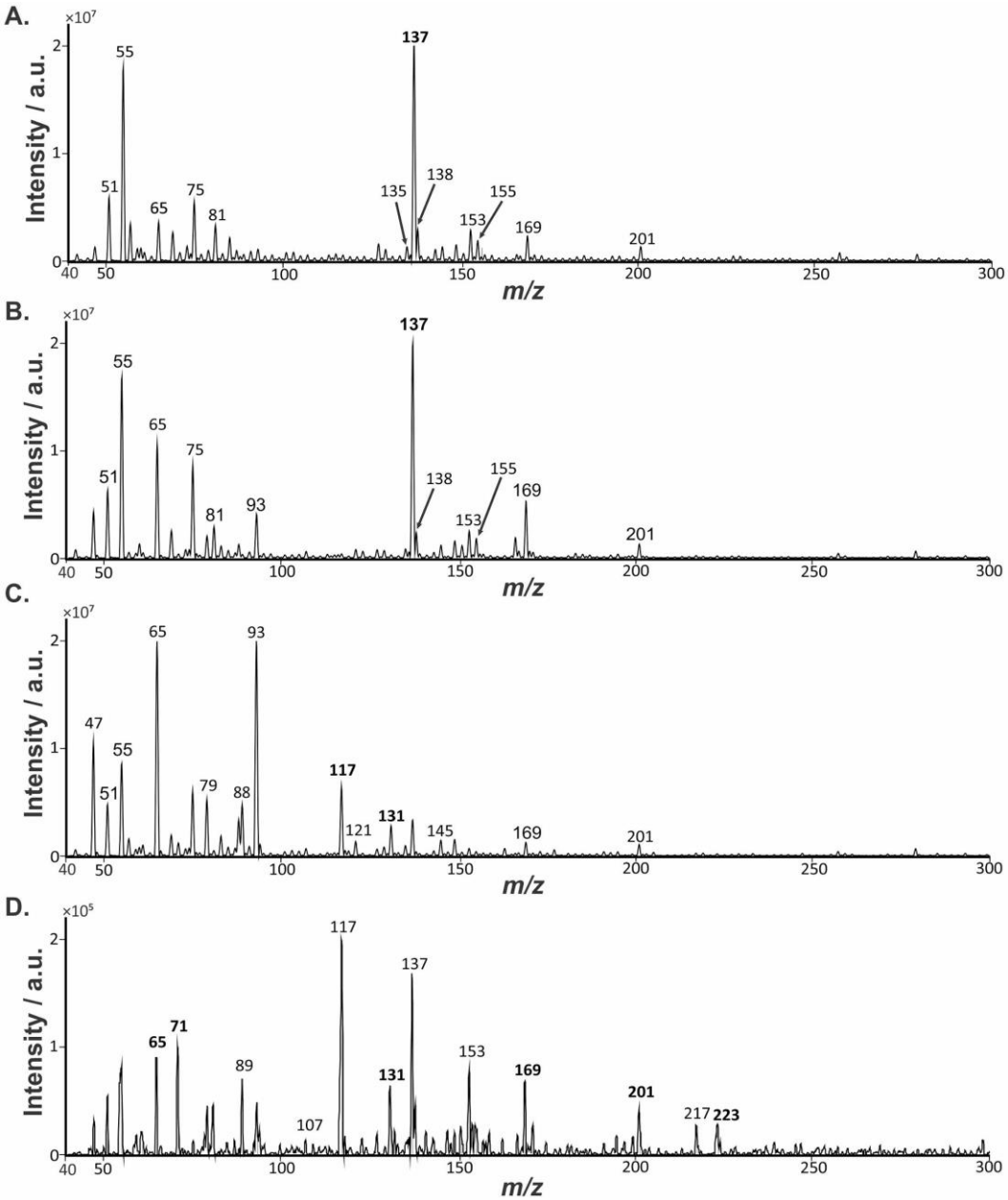

Figure S7. Analysis of VOCs by COME in conjunction with APCI-MS (*cf.* **Figure 1C,D**). Averaged spectra from 0.4 to 0.8 min for different samples. Ions (*m/z*) plotted in **Figure 3B** are indicated in bold. In the case of lemon drink, (A) *m/z* 73, 81, 135, 137 (pinene, myrcene, limonene, and terpinene), 138, 153, and 155 were observed. In the case of orange juice (B), *m/z* 81, 137 (limonene), 138, 153, and 155 were observed. In the case of apple flavor milk (C), *m/z* 88, 117 (ethyl butyrate), 121, 131 (2-methylbutyl acetate), and 145 were observed. In the case of the stored wastewater sample (D), we could not identify the signals. However, some ions (*m/z* 55, 61, 71, 75, 79, 81, 89, 93, 107, 117, 131, 137, 153, 169, 191, 217, 223, and 239) were observed in the spectrum of this sample.

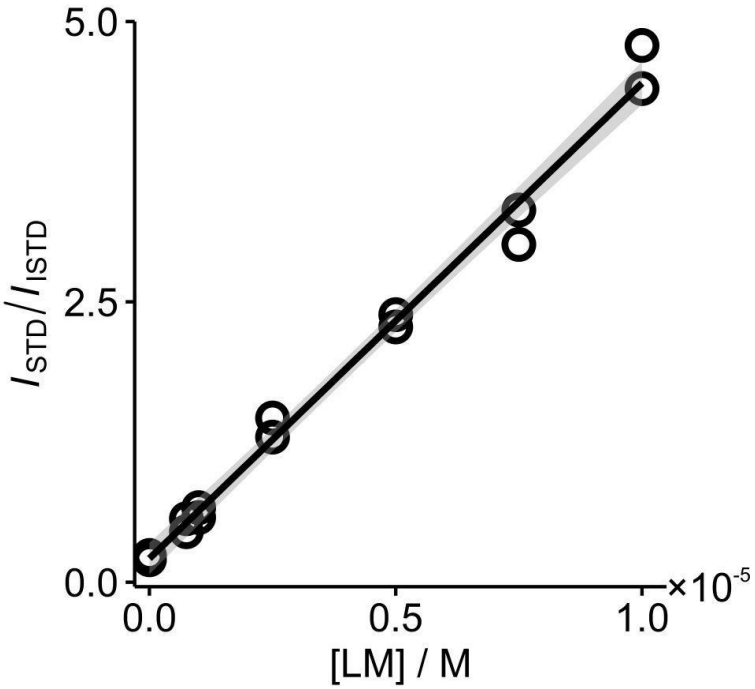

Figure S8. Calibration plot for LM (*m/z* 137→81) analyzed by APCI-MS method in MRM (positive mode). HA (5×10^{-6} M, *m/z* 145→61) was used as an internal standard. Average signal intensities within 24-48 s intervals were used for plotting. Fitted function: (I_{STD}/I_{ISTD} = $[(4.24\pm0.12)\times10^{5}]C_{STD}$+ (0.211±0.064), R^2 = 0.989). Conditions: catalase activity, 300 U mL^{-1}; hydrogen peroxide pumping time, 20 s; injection time, 60 s; nebulizing gas flow rate, 2.0 L min^{-1}; drying gas flow rate, 10 L min^{-1}; heated block temperature, 250 °C; ionization, APCI positive-ion mode; corona needle voltage, 4.5 kV.

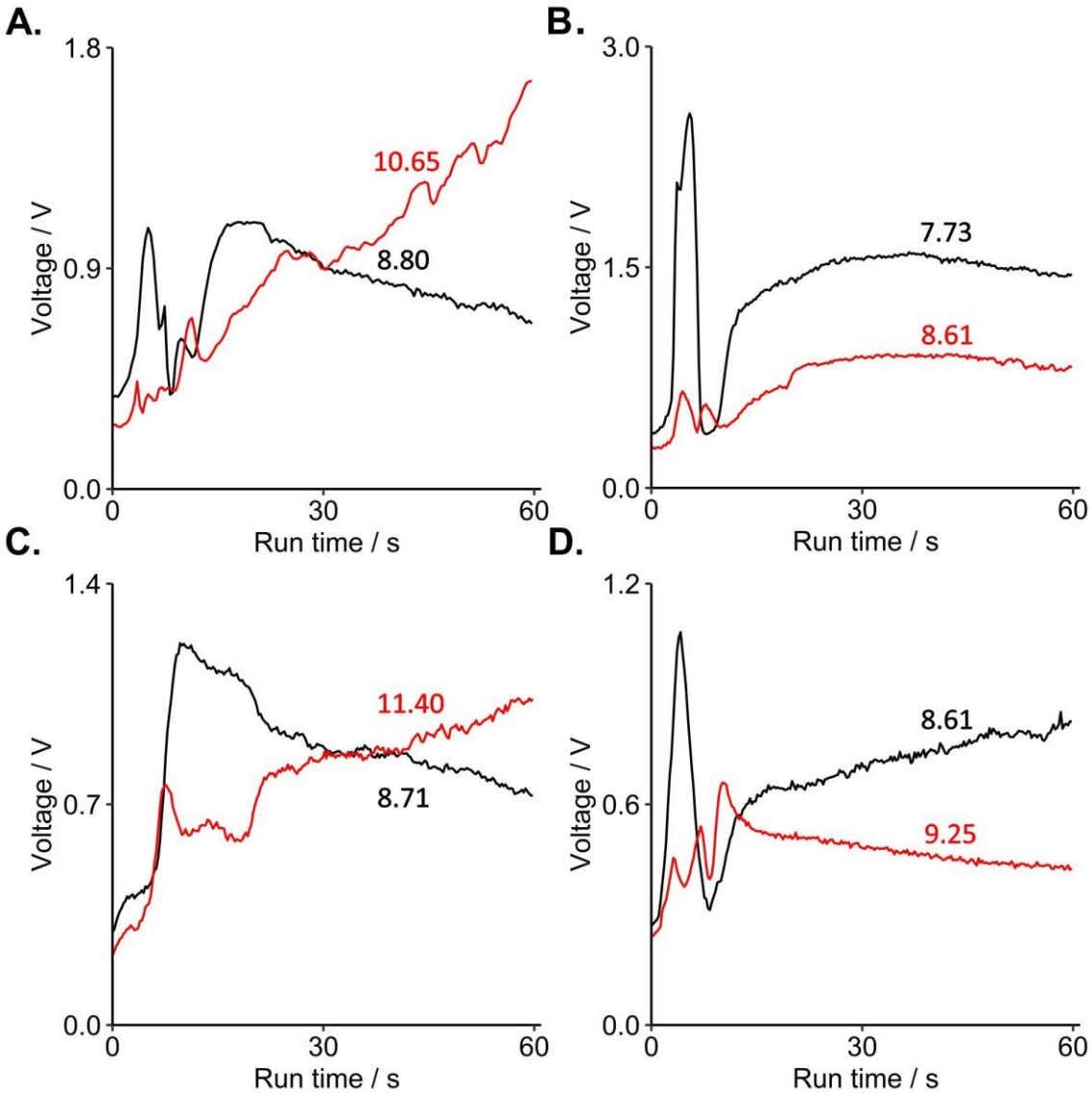

Figure S9. Analysis of VOCs by COME in conjunction with APCI-IMS (*cf.* **Figure 1E,F**). The ion currents are related to **Figure 3C**. Fluctuations observed from 0 to 20 s are due to uncontrolled flow of gas passing through the IMS. Panels: (A) American whiskey; (B) Scotch whiskey; (C) strawberry-flavored yogurt; (D) urine.

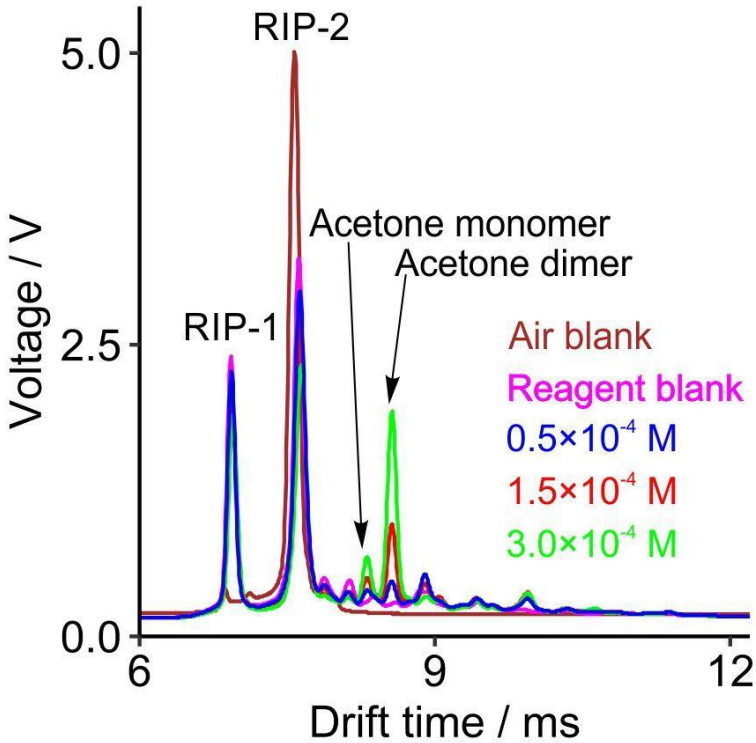

Figure S10. Analysis of acetone by COME in conjunction with APCI-IMS (the spectra from 20-60 s interval were averaged). Conditions: catalase activity, 300 U mL^{-1}; hydrogen peroxide pumping time, 20 s; injection time, 60 s; drift voltage polarity, positive; injection pulse width, 150 μs; drift voltage, 240 V; drift gas flow rate, 157 mL min^{-1}; recording time, 60 s; repetition rate, 20 ms; number of averaged spectra, 12; blocking voltage, 120 V; drift tube temperature, 70 °C.

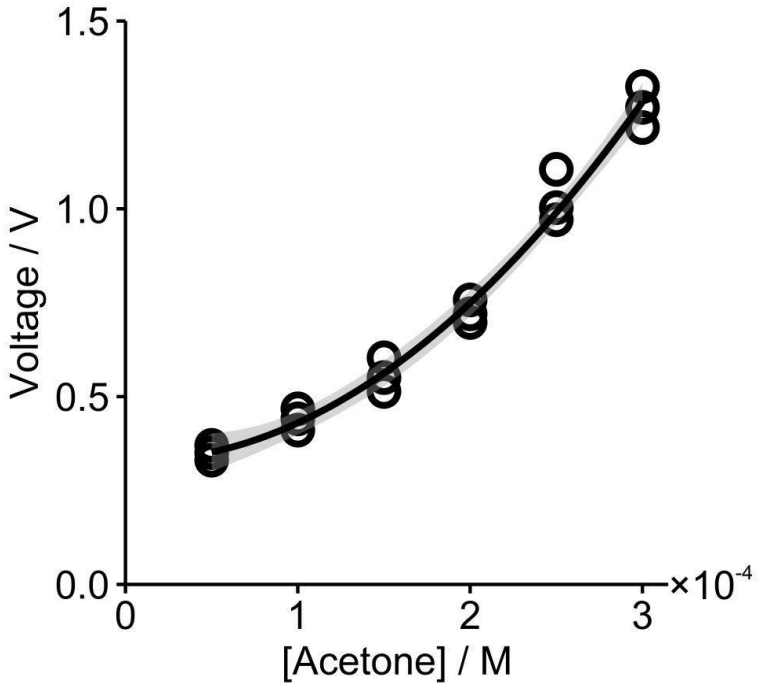

Figure S11. Calibration plot for dimer of acetone analyzed by APCI-IMS. EICs have been used (drift time = 8.55 ms, recording time = 20 to 60 s). Calibration equation: $U_{\text{acetone-dimer}} = [(1.08\pm0.17)\times10^{7})C_{\text{acetone-dimer}}^{2} + (-58.3\pm612.1)C_{\text{acetone-dimer}} + (0.327\pm0.046)$, $R^2 = 0.984$. Conditions: catalase activity, 300 U mL^{-1}; hydrogen peroxide pumping time, 20 s; injection time, 60 s; drift voltage polarity, positive; injection pulse width, 150 μs; drift voltage, 240 V; drift gas flow rate, 157 mL min^{-1}; recording time, 60 s; repetition rate, 20 ms; number of averaged spectra, 12; blocking voltage, 120 V; drift tube temperature, 70 °C.

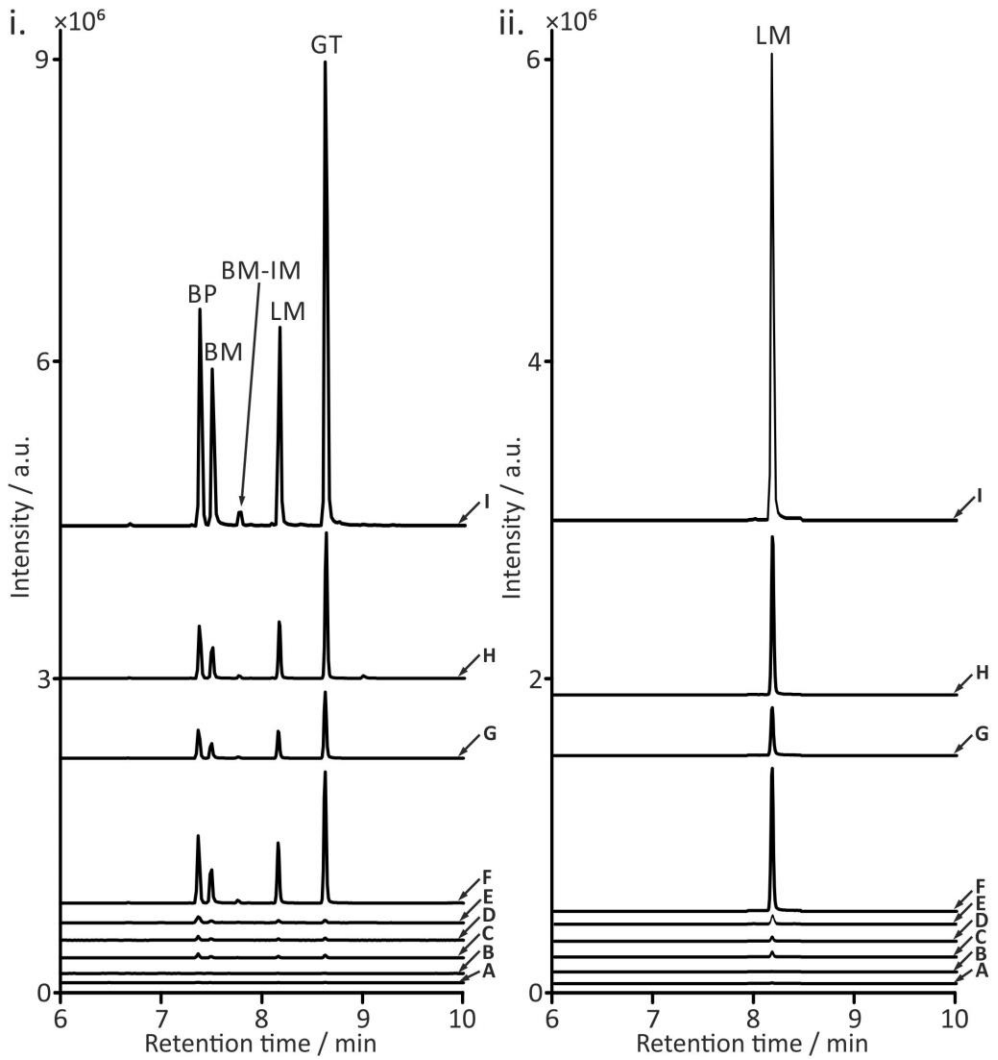

Figure S12. Comparison of chromatograms obtained by different sampling/extraction techniques used in conjunction with GC-MS. Concentrations of the four test compounds (BP, BM, LM, and GT) were 1×10^{-8} M. SIM mode was used for data acquisition as follows: start acquisition at 4.00 min, *m/z* 43, 56, 69, 93, and 121; from 7.92 min, *m/z* 68, and 93; and finally from 8.45 to 16.00 min, *m/z* 93, 121, and 136. EICs at the *m/z* = 93 (i.) and 68 (ii.) are displayed. Labels: (A) HS-SDME (room temp., 15 min); (B) HS-SDME (40 °C, 5 min); (C) HS (room temp., 15 min); (D) HS (40 °C, 5 min); (E) COME; (F) HS-SPME (room temp., 15 min); (G) HS-SPME (40 °C, 5 min); (H) COME + SPME (room temp., 5 min); and (I) COME + SPME (room temp., 15 min). One replicate out of two is shown. SDME droplet volume: 1.0 µL of ethanol. SPME fiber with 100-µm polydimethylsiloxane coating was used. Headspace injection volume: 700 µL. Sample mixture in headspace vial was 3 mL (50 mM potassium dihydrogen phosphate buffer adjusted to pH 7.0 with sodium hydroxide). In the case of COME + HS-SPME (catalase COME coupled with SPME) 150 U mL^{-1} catalase was used to avoid over pressurization of headspace vial (*i.e.* 2210 µL 50 mM potassium dihydrogen phosphate buffer adjusted to pH 7.0 with sodium hydroxide, 30 µL standard solution (1×10^{-6} M), 180 µL 2500 U mL^{-1} catalase, and 580 µL ~ 35% hydrogen peroxide solution). The SPME fiber was inserted into the catalase COME headspace vial after a 1-min delay to prevent the fiber from touching the foam.

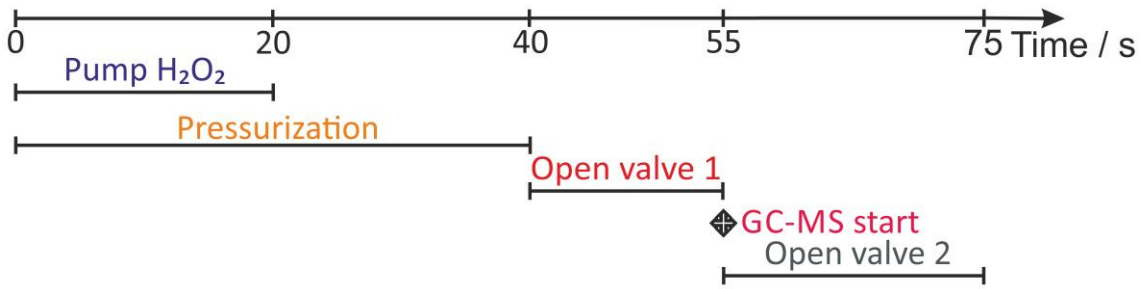

Figure S13. Time program for COME with potato pulp and manganese(IV) dioxide.

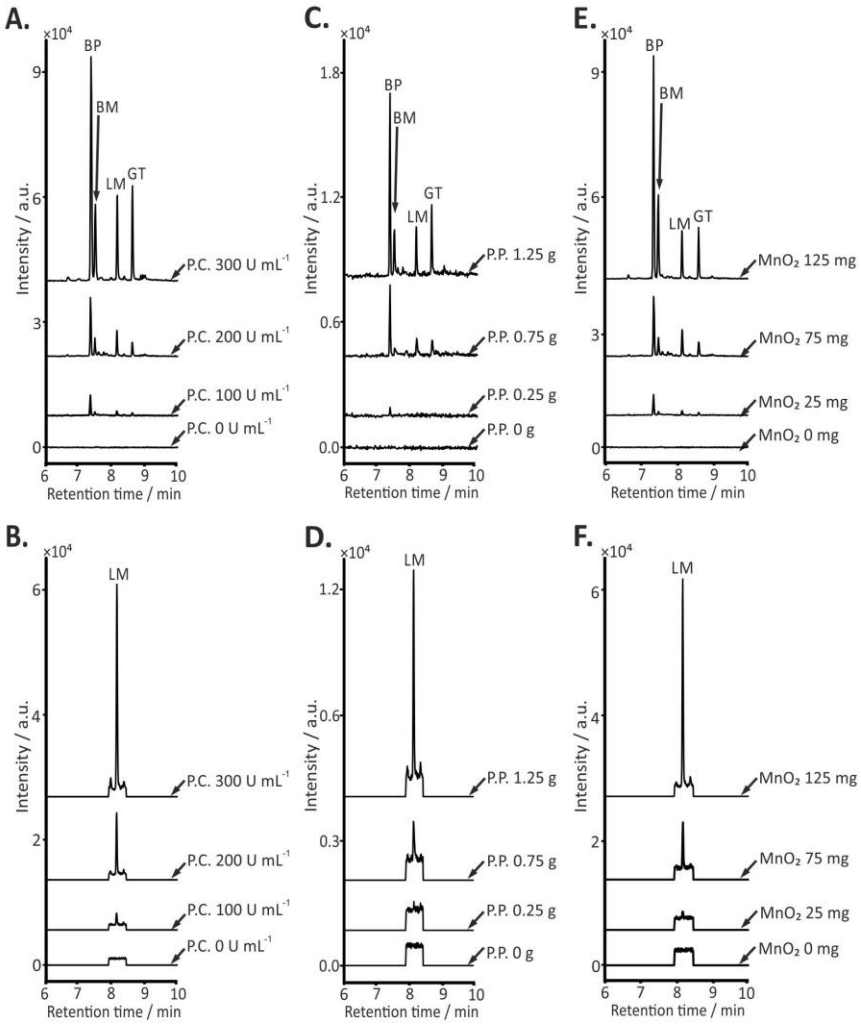

Figure S14. Comparison of COME with pure catalase (A, *m/z* 93; B, *m/z* 68), crude potato pulp (C, *m/z* 93; D, *m/z* 68), and manganese(IV) dioxide (E, *m/z* 93; F, *m/z* 68). Concentrations of the four test compounds (BP, BM, LM, and GT) were 1×10^{-8} M. SIM mode was used for data acquisition as follows: start acquisition at 4.00 min, *m/z* 43,56, 69, 93, and 121; from 7.92 min, *m/z* 68, and 93; and finally from 8.45 to 16.00 min, *m/z* 93, 121, and 136. EICs at the *m/z* 93 and *m/z* 68 are displayed. One replicate out of two is shown. Chromatogram labels: P.C., pure catalase; P.P., potato pulp; MnO_2, manganese(IV) dioxide. In the case of pure catalase COME, the final mixture volume was 3 mL. Final extraction mixture in P.C. variant: 2030-2270 µL buffer (50 mM potassium dihydrogen phosphate buffer adjusted to pH 7.0 with sodium hydroxide); 30 µL of standard solution (1×10^{-6} M); 360 µL catalase stock solution (2500 U mL^{-1}); 580 µL ~ 35% hydrogen peroxide solution. Final extraction mixture in P.P. variant: 2390 µL buffer (50 mM potassium dihydrogen phosphate buffer adjusted to pH 7.0 with sodium hydroxide); 0-1.25 g potato pulp; 30 µL of standard solution (1×10^{-6} M); 580 µL hydrogen peroxide (35%). Final extraction mixture in MnO_2 variant: 2390 µL buffer (50 mM potassium dihydrogen phosphate buffer adjusted to pH 7.0 with sodium hydroxide); 0-125 mg manganese(IV) dioxide; 30 µL of standard solution (1×10^{-6} M); 580 µL ~ 35% hydrogen peroxide solution. In the case of crude potato pulp and manganese(IV) dioxide COME, the final mixture volume was ~ 3 mL. Further, in the case of crude potato pulp and manganese(IV) dioxide COME, the extraction chamber was pressurized for 40 s before injection because bubbling was relatively slow (*cf.* **Figure S13**).

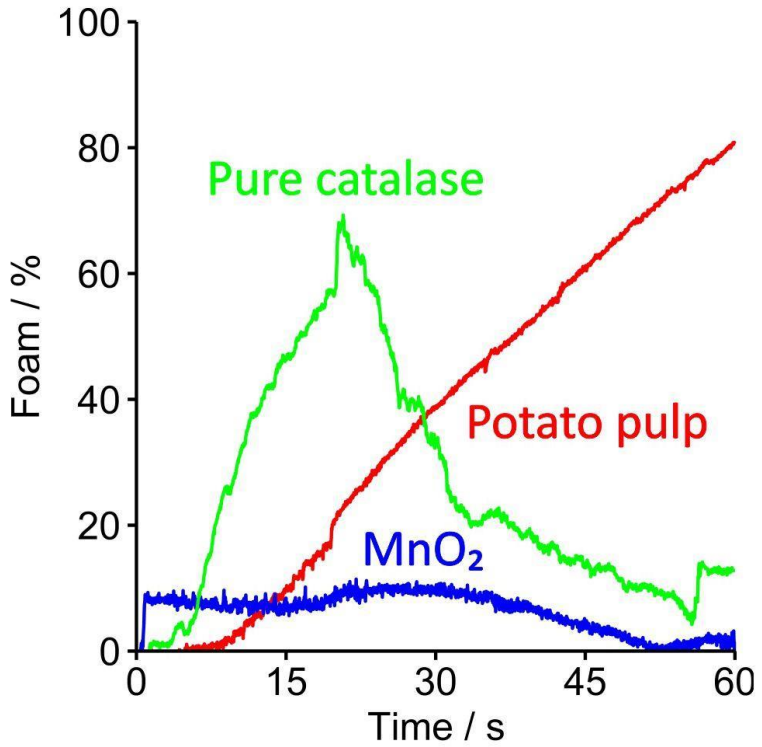

Figure S15. Comparison of foam profiles of pure catalase (green), potato pulp (red), and manganese(IV) dioxide (blue, MnO_2). In the case of pure catalase, 300 U mL^{-1} catalase concentration was used, and the final mixture volume was 3 mL [2060 µL buffer (50 mM potassium dihydrogen phosphate buffer adjusted to pH 7.0 with sodium hydroxide); 360 µL catalase stock solution (2500 U mL^{-1}); 580 µL ~ 35% hydrogen peroxide solution (6.73×10^{-3} mol)]. In the comparative experiments, 1.25 g of potato pulp or 125 mg of manganese(IV) dioxide were used per 3 mL of the mixture comprising 2420 µL buffer and 580 µL ~ 35% hydrogen peroxide solution (6.73×10^{-3} mol). In the case of potato pulp and manganese(IV) dioxide, the extraction chamber was pressurized for 40 s before injection because bubbling was relatively slow.

COMPUTER CODES

Arduino code for MCB (trigger through clap sound and push button; GC-MS)

```
const int pump = 12;    // Assigning a pin to the peristaltic pump.
const int NC = 13;      // Assigning a pin to valve 2.
const int NO = 14;      // Assigning a pin to valve 1.
const int GC = 16;      // Assigning a pin to the GC data acquisition.
const int button = 4;   // Assigning a pin to the push button to start extraction.
const int sensor = A0;  // Assigning a pin to the sound sensor to start extraction by clap.
const int trigger = 5;  // Assigning a pin to trigger the SBC to start the camera.
const int hold = 15;    // Assigning a pin to receive a digital signal from the SBC.

int Status1 = 0;
int Status2 = 0;
int Signal = 0;
int i1 = 0;

void setup() {          // Assigning functions to the pins.
  Serial.begin(115200);
  pinMode(pump, OUTPUT);
  pinMode(NC, OUTPUT);
  pinMode(NO, OUTPUT);
  pinMode(GC, OUTPUT);
  pinMode(trigger,OUTPUT);
  pinMode(button, INPUT);
  pinMode(sensor, INPUT);
  pinMode(hold, INPUT);
  digitalWrite(pump,0);
  digitalWrite(NC,0);
  digitalWrite(NO,0);
  digitalWrite(GC,0);
}
```

```
void loop() {                          // This loop is used to trigger extraction.
  Status1 = digitalRead(button);
  Status2 = analogRead(sensor);
  if (Status1==1||Status2>=1000){      // This loop is used to trigger the extraction by push button or clap.
    digitalWrite(trigger,1);           // Start SBC camera.
    for(int i=0; i<20000; i++){        // This loop is used to pump hydrogen peroxide for 20 s.
      Signal = digitalRead(hold);
      digitalWrite(pump,1);            // Turn on the pump.
      digitalWrite(NO,1);              // Close valve 1
      digitalWrite(NC,0);              // Close valve 2
      delay(1);
      i1=20000-i;
      if(Signal==1){                   // Receive signal from SBC.
        digitalWrite(pump,0);          // Turn off the pump.
        i = 20001;
        if(i1>0){                      // Delay the release of gas for 20 s to pressure the extraction
chamber.
          delay(i1);
        }
      }
    }
    digitalWrite(NO,0);                // Open Valve 1.
    digitalWrite(NC,0);                // Close Valve 2.
    delay(15000);                      // Valve 1 is open for 15 s.
    digitalWrite(GC, 1);      // Trigger GC for data acquisition.
    delay(100);
    digitalWrite(GC,0);
    digitalWrite(NC,1);                // Open valve 2.
    digitalWrite(NO,1);                // Close valve 1.
    delay(20000);                      // Valve 2 is open for 20 s.
  }
  else{
    digitalWrite(pump,0);
    digitalWrite(NC,0);
    digitalWrite(NO,0);
    digitalWrite(GC,0);
    digitalWrite(trigger,0);
```

```
    }
  }
// End of code.
```

Arduino code for MCB (trigger through smartphone; GC-MS)

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266mDNS.h>
#include <ESP8266WebServer.h>
ESP8266WebServer server(80);
const char* ssid="XXXXXXX";          // Please insert the username.
const char* password="XXXXXXXX";     // Please insert the password.
const char* myssid="mahesh network";
const int pump = 12;                 // Assigning a pin to the peristaltic pump.
const int NC = 13;                   // Assigning a pin to the valve 2.
const int NO = 14;                   // Assigning a pin to the valve 1.
const int GC = 16;                   // Assigning a pin to the GC data acquisition.
const int trigger = 5;               // Assigning a pin to trigger the SBC to start the camera.
const int hold=15;                   // Assigning a pin to receive a digital signal from the SBC.
int Signal = 0;
int Status1 = 0;
int Status2 = 0;
IPAddress ip(XXX,XXX,X,XXX);         // Please insert the IP address.
IPAddress gateway(192,168,11,1);
IPAddress subnet(255,255,255,0);

void setup() {                       // Assigning functions to the pins and setup WiFi.
 Serial.begin(115200);
  pinMode(pump, OUTPUT);
  pinMode(NC, OUTPUT);
  pinMode(NO, OUTPUT);
  pinMode(GC, OUTPUT);
  pinMode(trigger,OUTPUT);
```

```
  digitalWrite(pump,0);
  digitalWrite(NC,0);
  digitalWrite(NO,0);
  digitalWrite(GC,0);
  WiFi.mode(WIFI_AP_STA);
  WiFi.begin(ssid,password);
  while(WiFi.status()!=WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.print("IP Address:");
  Serial.println(WiFi.localIP());
  server.on("/",[](){server.send(200,"text/pain","Hello Mahesh! I am online");});
  server.on("/s", s);
  server.begin();
  WiFi.softAPConfig(ip,gateway,subnet);
  WiFi.softAP(myssid, password);
}

void loop() {                          // This loop provides web server control.
  server.handleClient();
 }

void s(){                              // This loop is used to trigger extraction with a mobile phone.
 digitalWrite(trigger,1);              // Start SBC camera.
    for(int i=0; i<20000; i++){        // This loop is used to pump hydrogen peroxide for 20 s.
      Signal = digitalRead(hold);
      digitalWrite(pump,1);            // Turn on the pump.
      digitalWrite(NO,1);              // Close valve 1.
      digitalWrite(NC,0);              // Close valve 2.
      delay(1);
      i1=20000-i;
      if(Signal==1){                   // Receive signal from SBC.
        digitalWrite(pump,0);          // Turn off the pump.
        i = 20001;
```

```
        if(i1>0){                   // Delay the release of gas for 20 s to pressure the extraction
chamber.
          delay(i1);
        }
      }
    }
    digitalWrite(NO,0);             // Open Valve 1.
    digitalWrite(NC,0);             // Close Valve 2.
    delay(15000);                   // Valve 1 open for 15 s.
    digitalWrite(GC, 1);      // Trigger GC for data acquisition.
    delay(100);
    digitalWrite(GC,0);
    digitalWrite(NC,1);             // Open valve 2.
    digitalWrite(NO,1);             // Close valve 1.
    delay(20000);                   // Valve 2 open for 20 s.
}
// End of code.
```

Arduino code for MCB (trigger through clap sound and push button; APCI-MS and APCI-IMS)

```
const int pump = 12;                // Assigning a pin to the peristaltic pump.
const int NC = 13;                  // Assigning a pin to the valve 2.
const int NO = 14;                  // Assigning a pin to the valve 1.
const int QQQ = 16;                 // Assigning a pin to the MS data acquisition.
const int button = 4;               // Assigning a pin to the push button to start extraction.
const int sensor = A0;              // Assigning a pin to the sound sensor to start extraction by clap.
const int trigger = 5;              // Assigning a pin to trigger the SBC to start the camera.
const int hold = 15;                // Assigning a pin to receive a digital signal from the SBC.

int Status1 = 0;
int Status2 = 0;
int Signal = 0;
int i1=0;

void setup() {                      // Assigning functions to the pins.
  Serial.begin(115200);
```

```
  pinMode(pump, OUTPUT);
  pinMode(NC, OUTPUT);
  pinMode(NO, OUTPUT);
  pinMode(QQQ, OUTPUT);
  pinMode(trigger,OUTPUT);
  pinMode(button, INPUT);
  pinMode(sensor, INPUT);
  pinMode(hold, INPUT);
  digitalWrite(pump,0);
  digitalWrite(NC,0);
  digitalWrite(NO,0);
  digitalWrite(QQQ,0);
}

void loop() {                         // This loop is used to trigger extraction.
  Status1 = digitalRead(button);
  Status2 = analogRead(sensor);
  if (Status1==1||Status2>=1000){     // This loop condition is used to trigger the extraction by push button
or clap.
    digitalWrite(trigger,1);          // Start SBC camera.
    for(int i=0; i<20000; i++){       // This loop is used to pump hydrogen peroxide for 20 s.
      Signal = digitalRead(hold);
      digitalWrite(pump,1);           // Turn on the pump.
      digitalWrite(QQQ, 1);           // Trigger QQQ for data acquisition.
      digitalWrite(NO,0);             // Open valve 1.
      digitalWrite(NC,0);             // Close valve 2.
      delay(1);
      i1=20000-i;
      if(Signal==1){                  // Receive signal from SBC.
        digitalWrite(pump,0);               // Turn off the pump.
        i = 20001;
        if(i1>0){                           // Delay the release of gas for 20 s to pressure the extraction
chamber.
            delay(i1);
        }
      }
    }
```

```
    digitalWrite(NO,0);             // Open Valve 1.
    digitalWrite(NC,0);             // Close Valve 2.
    delay(40000);                   // Valve 1 is open for 60 s (hydrogen peroxide injection time 20 s and
delay 40 s).
    digitalWrite(NC,1);             // Open valve 2.
    digitalWrite(NO,1);             // Close valve 1.
    delay(20000);                   // Valve 2 is open for 20 s.
  }
  else{
    digitalWrite(pump,0);
    digitalWrite(NC,0);
    digitalWrite(NO,0);
    digitalWrite(GC,0);
    digitalWrite(trigger,0);
    }
  }
// End of code.
```

Arduino code for MCB (trigger through smartphone; APCI-MS and APCI-IMS)

```
#include <ESP8266WiFi.h>
#include <WiFiClient.h>
#include <ESP8266WiFiMulti.h>
#include <ESP8266mDNS.h>
#include <ESP8266WebServer.h>
ESP8266WebServer server(80);
const char* ssid="XXXXXXX";         // Please insert the username.
const char* password="XXXXXXXX";    // Please insert the password.
const char* myssid="mahesh network";
const int pump = 12;                // Assigning a pin to the peristaltic pump.
const int NC = 13;                  // Assigning a pin to the valve 2.
const int NO = 14;                  // Assigning a pin to the valve 1.
const int QQQ = 16;                 // Assigning a pin to the data acquisition in QQQ.
const int trigger = 5;              // Assigning a pin to trigger the SBC to start the camera.
const int hold = 15;                // Assigning a pin to receive a digital signal from the SBC.
int Signal = 0
```

```
int Status1 = 0;
int Status2 = 0;
IPAddress ip(XXX,XXX,X,XXX);        // Please insert the IP address.
IPAddress gateway(192,168,11,1);
IPAddress subnet(255,255,255,0);

void setup() {                      // Assigning functions to the pins and setup WiFi.
 Serial.begin(115200);
  pinMode(pump, OUTPUT);
  pinMode(NC, OUTPUT);
  pinMode(NO, OUTPUT);
  pinMode(QQQ, OUTPUT);
  pinMode(trigger, OUTPUT);
  digitalWrite(pump,0);
  digitalWrite(NC,0);
  digitalWrite(NO,0);
  digitalWrite(QQQ,0);
  WiFi.mode(WIFI_AP_STA);
  WiFi.begin(ssid,password);
  while(WiFi.status()!=WL_CONNECTED)
  {
    Serial.print(".");
    delay(500);
  }
  Serial.println("");
  Serial.print("IP Address:");
  Serial.println(WiFi.localIP());
  server.on("/",[](){server.send(200,"text/pain","Hello Mahesh! I am online");});
  server.on("/s", s);
  server.begin();
  WiFi.softAPConfig(ip,gateway,subnet);
  WiFi.softAP(myssid, password);
}

void loop() {                       // This loop provides web server control.
  server.handleClient();
 }
```

```
void s(){                              // This loop is used to trigger extraction with a mobile phone.
 digitalWrite(trigger,1);              // Start SBC camera.
    for(int i=0; i<20000; i++){        // This loop is used to pump hydrogen peroxide for 20 s.
      Signal = digitalRead(hold);
      digitalWrite(pump,1);            // Turn on the pump.
      digitalWrite(QQQ, 1);            // Trigger QQQ for data acquisition.
      digitalWrite(NO,0);              // Open valve 1.
      digitalWrite(NC,0);              // Close valve 2.
      delay(1);
      i1=20000-i;
      if(Signal==1){                   // Receive signal from SBC.
        digitalWrite(pump,0);          // Turn off the pump.
        i = 20001;
        if(i1>0){                      // Delay the release of gas for 20 s to pressure the extraction
chamber..
          delay(i1);
        }
      }
    }
    digitalWrite(NO,0);                // Open Valve 1.
    digitalWrite(NC,0);                // Close Valve 2.
    delay(40000);                      // Valve 1 is open for 60 s (hydrogen peroxide injection time 20 s and
delay 40 s).
    digitalWrite(NC,1);                // Open valve 2.
    digitalWrite(NO,1);                // Close valve 1.
    delay(20000);                      // Valve 2 is open for 20 s.
}
// End of code.
```

Python code for SBC (reaction monitoring)
- parts adapted from: https://docs.opencv.org/master/df/d9d/tutorial_py_colorspaces.html

```
import cv2 as cv                       # Required libraries are imported.
import numpy as np
import DateTime
```

```
import csv
import time
import os.path
import dropbox as dp
from periphery import*
class TransferData:                  # Function used for data transfer to the Dropbox.
   def __init__(self, access_token):
       self.access_token = access_token
   def upload_file(self, file_from, file_to):
       """upload a file to Dropbox using API v2
       """
       dbx = dp.Dropbox(self.access_token)

       with open(file_from, 'rb') as f:
           dbx.files_upload(f.read(), file_to)
gpio6=GPIO(6,"in")                   # Pin 6 is used to receive the signal from the MCB.
gpio16=GPIO(16,"out")                # Pin 16 is used to send signal to the MCB.
while True:                          # This loop is used to trigger the camera and acquire data from it.
   value=gpio6.read()
   if value==True:
       capture_duration = 60         # Time of capture video.
       cap=cv.VideoCapture(0)
       cap.set(cv.CAP_PROP_FRAME_WIDTH,640)
       cap.set(cv.CAP_PROP_FRAME_HEIGHT,480)
       start_time = time.time()
       save_path = '/home/mendel/Desktop'       # Path to save data.
       name_of_file = 'Data' + time.strftime("%Y-%H-%M")
       filename = os.path.join(save_path, name_of_file + ".csv")
       with open(filename, "a", newline='') as f:
           writer = csv.writer(f, delimiter=",")
           writer.writerow(['Time', 'Intensity'])
           while (int(time.time() - start_time) < capture_duration):
               ret, frame = cap.read()
               if ret == True:      # This loop is used to capture the pixel values in the image.
                   hsv = cv.cvtColor(frame, cv.COLOR_BGR2HSV)
                   l_b = np.array([0, 0,165])
                   u_b = np.array([120,255, 255])
```

```
                mask = cv.inRange(frame, l_b, u_b)
                real = cv.bitwise_and(frame, frame, mask=mask) #  This  bitwise_and  operator  is  used
    optionally to display color images after applying the mask.
                real = cv.resize(real, (200, 200))
                ims = cv.resize(mask, (200, 200))
                x = ims
                value = x.sum()
                Signal = gpio16.write()
                if (value < 25000):          # This loop is used to send signal to the MCB.
                       Signal.write(True)
                else:
                  signal.write(False)
                now = float(time.time()-start_time)
                now1=float("{:.2f}".format(now))
                print(now1,value)
                writer.writerow([now1, value])
            else:
                break
      cap.release()
      cv.destroyAllWindows()
 access_token ='XXXXXXXXXXXXX'                   # Please insert your access code.
      transferData = TransferData(access_token)
      print (filename)
      file_from = filename
      filetarget = '/Opencvdata/'+name_of_file+".CSV"
      file_to = filetarget                       # The full path to upload the file.
      print(file_to)
      transferData.upload_file(file_from, file_to)
  elif value==False:                             # Camera stopped to capture the video.
      print("Camera off")
# End of code.
```